Expert Systems With Applications 212 (2023) 118712

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Deep learning model for doors detection: A contribution for context-awareness recognition of patients with Parkinson's disease

Helena R. Gonçalves [a,*], Cristina P. Santos [b]

[a] University of Minho, Center for MicroElectroMechanical, Portugal
[b] University of Minho, Center for MicroElectroMechanical, Portugal

ARTICLE INFO

Keywords:
Object detection
Deep-learning
RPi
Parkinson's disease

ABSTRACT

Freezing of gait (FoG) is one of the most disabling motor symptoms in Parkinson's disease, which is described as a symptom where walking is interrupted by a brief, episodic absence, or marked reduction, of forward progression despite the intention to continue walking. Although FoG causes are multifaceted, they often occur in response of environment triggers, as turnings and passing through narrow spaces such as a doorway. This symptom appears to be overcome using external sensory cues. The recognition of such environments has consequently become a pertinent issue for PD-affected community. This study aimed to implement a real-time DL-based door detection model to be integrated into a wearable biofeedback device for delivering on-demand proprioceptive cues. It was used transfer-learning concepts to train a MobileNet-SSD in TF environment. The model was then integrated in a RPi being converted to a faster and lighter computing power model using TensorFlow Lite settings. Model performance showed a considerable precision of 97,2%, recall of 78,9% and a good F1-score of 0,869. In real-time testing with the wearable device, DL-model showed to be temporally efficient (~2.87 fps) to detect with accuracy doors over real-life scenarios. Future work will include the integration of sensory cues with the developed model in the wearable biofeedback device aiming to validate the final solution with end-users.

1. Introduction

Freezing of gait (FoG) is one of the most disabling motor symptoms in Parkinson's disease (PD), which affects approximately half of the patients in advanced illness stages (Lewis & Barker, 2009; Pham et al., 2017; Sweeney et al., 2019). FoG is described as a symptom where walking is interrupted by a brief, episodic absence, or marked reduction, of forward progression despite the intention to continue walking (Ehgoetz Martens, Pieruccini-Faria, & Almeida, 2013; Lewis & Barker, 2009). A pathological model of FoG suggests that with additional demands of sensory overload from visual or other stimuli leads to an overactivation of typical basal-ganglia pathways leading to an inhibition of movement and occurrence of freezing episodes (Lewis & Barker, 2009). Although FoG causes are multifaceted, they often occur in response of environment triggers, as turnings and passing through narrow spaces such as a doorway (Gómez-Jordana, Stafford, Peper, & Craig, 2018; Lewis & Barker, 2009; Pham et al., 2017).

Most common situations where FoG occurs involves a change in the visual environment that might require integration of visual and proprioceptive information leading to a sensory overload and a disruption in basal ganglia pathways (Cowie, Limousin, Peters, Hariz, & Day, 2012; Gómez-Jordana et al., 2018; Lewis & Barker, 2009). This disruption appears to be overcome using external sensory cues (visual, auditory and/or vibrotactile), which may help basal ganglia circuits to shifts to healthy spared neural motor pathways (Ehgoetz Martens et al., 2013). External cues can provide spatial and temporal information by indicating where or when patients should place their feet while walking, especially in front of such FoG-trigger environments (Delgado-Alvarado et al., 2019; Ehgoetz Martens et al., 2013; Ginis, Nackaerts, Nieuwboer, & Heremans, 2018; Muthukrishnan, Abbas, Shill, & Krishnamurthi, 2019). It is hypothesized that patients in the presence of external cues exclusively focus on walking by promoting a goal-oriented gait,

Peer review under responsibility of Submissions with the production note 'Please add the Reproducibility Badge for this item' the Badge and the following footnote to be added:The code (and data) in this article has been certified as Reproducible by the CodeOcean: https://codeocean.com. More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physicalsciencesandengineering/computerscience/journals.

* Corresponding author at: Center for MicroElectroMechanical, University of Minho – Azurém Campus, 4800-058 Guimarães, Portugal.
E-mail addresses: id7609@alunos.uminho.pt (H.R. Gonçalves), cristina@dei.uminho.pt (C.P. Santos).

https://doi.org/10.1016/j.eswa.2022.118712
Received 8 July 2022; Received in revised form 17 August 2022; Accepted 25 August 2022
Available online 30 August 2022

particularly in the specific FoG trigger situations, such as passing through doors or turnings (Chong & -Hyun Lee, K., Morgan, J., & Mehta, S., 2011). The recognition of such environments has consequently become a pertinent issue for the PD-affected community (context-awareness recognition). Indeed, the recognition of turning to provide sensory cues has already been addressed by (Mancini, Smulders, Harker, Stuart, & Nutt, 2018) and (Harrington et al., 2016), where gait improvements have been observed with their developed wearable biofeedback device. However, to the best knowledge of the authors, no other study focused on door detection in the PD field, remaining an unaddressed and important challenge for the targeted scientific community. These scenarios have been so important in PD motor symptoms management, that curiously, virtual doors have been used by (Gómez-Jordana et al., 2018; Yamagami et al., 2020) to produce a new framework to assess and train patients in this specific FoG-trigger scenario. Thus, the detection of doors comprises a meaningful step in advancing the development of devices for motor assistance or rehabilitation in PD, which led us to develop a camera-based system for door detection to be integrated into a wearable biofeedback device. Besides our solution bringing the innovative capability of door detection in devices for motor assistance or rehabilitation in PD, it will complement current approaches of biofeedback for turnings (Harrington et al., 2016; Mancini et al., 2018) or postural stability (Delgado-Alvarado et al., 2019).

Recent advances in sensing technology and the appearance of low-cost devices such as miniaturized camera modules have made the collection of images more feasible and affordable, besides of being easily integrated into a wearable devices (Ioannidou, Chatzilari, Nikolopoulos, & Kompatsiaris, 2017; Voulodimos, Doulamis, Doulamis, & Protopapadakis, 2018). The increasing abundance of captured images encouraged the research community to exploit this richer content for addressing several computer vision problems related to understanding scenarios for object classification, detection, and tracking. Over the last years, deep learning (DL) methods have been shown to outperform previous state-of-the-art (SoA) machine learning techniques in several fields, with computer vision being one of the most prominent cases (Ioannidou et al., 2017; Voulodimos et al., 2018; Zhao, Zheng, Xu, & Wu, 2018). DL uses computational models of multiple processing layers to learn and represent data with numerous levels of abstraction (Voulodimos et al., 2018). Its use for computer vision can be categorized into various categories: generation, segmentation, classification and detection of both videos and images (Krishna Sai & Sasikala, 2019). Object detection in DL consists in finding the position of an object and labelling (*e.g.,* a door), which solves the described problem of detecting doors using a camera-based system (Voulodimos et al., 2018).

Object detection using computer vision combined with DL has been intensively explored on last years for different areas, leading google brains to create an open-source and free software library for machine learning and DL, named TensorFlow (TF). Latest improvements in object detection driven researchers to introduce in TF a robust Object Detection API tool which enables the use of pre-trained models and construct, adapt and train new models. In most of the cases, training a complete convolutional network from scratch is time-consuming and needs massive datasets, as ImageNet dataset (Amit, Felzenszwalb, & Girshick, 2020). This problem was solved by using the power of transfer learning with a pre-trained model using the TF Object Detection API (Adrian Rosebrock, 2017). Doors' detection using DL has already been addressed for indoor robot navigation with social, assistance or domestic applications, as more detailed in section 2 (Banerjee, Long, Du, Polido, Feng, Atkeson, Gennert, & Padir, 2015; Borgsen, Schöpfer, Ziegler, & Wachsmuth, 2014; Chen, Qu, Zhou, Weng, Wang, & Fu, 2014; Dai et al., 2013; Derry & Argall, 2013; Fernández-Caramés, Moreno, Curto, Rodríguez-Aragón, & Serrano, 2014; He & Zhu, 2017; Kakillioglu, Ozcan, & Velipasalar, 2016; Lecrosnier et al., 2021; Llopart, Ravn, & Andersen, 2017; Othman & Rad, 2020; Quintana, Prieto, Adán, & Bosché, 2018; Ramoa, Alexandre, & Mogo, 2020; Sekkal, Pasteau, Babel, Brun, & Leplumey, 2013; Shalaby, Salem, Khamis, & Melgani, 2014; Spournias, Antonopoulos, Keramidas, Voros, & Stojanovic, 2020; Tian, Yang, Yi, & Arditi, 2013; Yuan, Hashim, Zaki, & Huddin, 2016). However, these studies did not address PD problem, a contribution introduced by this work.

This manuscript presents a DL-based object detection model, implemented in a Raspberry Pi board (RPi) and using an RPi camera module. It is expected to contribute for PD scope, aiming to develop a wearable biofeedback device for delivering proprioceptive cues by real-time detection of patients' environment using DL techniques. This work describes a step advance on high-tech to provide context-awareness biofeedback on PD. Our research addresses two main contributions: (1) end-to-end custom object (door) detection using transfer-learning from TF Object Detection API; and (2) implementation of trained object detection model in a RPi by conversion of TF-based model to TensorFlow Lite (TF-Lite) version.

2. Related work: Computer vision for door detection

Door detection approaches have been extensively developed for indoor robot navigation with social, assistance or domestic applications (Banerjee et al., 2015; Borgsen et al., 2014; Chen et al., 2014; Dai et al., 2013; Derry & Argall, 2013; Fernández-Caramés et al., 2014; He & Zhu, 2017; Kakillioglu et al., 2016; Lecrosnier et al., 2021; Llopart et al., 2017; Othman & Rad, 2020; Quintana et al., 2018; Ramoa et al., 2020; Sekkal et al., 2013; Shalaby et al., 2014; Spournias et al., 2020; Tian et al., 2013; Yuan et al., 2016). Robotic wheelchairs, humanoids, or systems for aid persons with visual impairments were other target fields of research in door detection algorithms (Derry & Argall, 2013; He & Zhu, 2017; Lecrosnier et al., 2021; Llopart et al., 2017; Othman & Rad, 2020; Ramoa et al., 2020; Shalaby et al., 2014; Tian et al., 2013). Table 1 summarizes the *state-of-the-art* survey in computer vision-based systems for door detection.

First approaches considered the geometric features of a door, being assumed as parallelogram. These lines were extracted using Hough transformation and Canny Edge algorithms, being subsequently accomplished a matching between the detected lines and corners (Derry & Argall, 2013; Hensler, Blaich, & Bittel, 2010; Shalaby et al., 2014; Tian, Yang, & Arditi, 2010).The use of these approaches demands common door features, being not adaptable for different shapes, colors, or frames. Besides, most of these studies were tested in controlled and pre-recognized environments.

Fuzzy logic was explored by Munoz-Salinas et al. (Munoz-Salinas, Aguirre, Garcia-Silvente, & Gonzalez, 2004) to analyze relationships between the segments identified by a doorframe model-based algorithm by using Hough Transform. The system successfully identified typical doors in different environments in real-time and regardless the perspective of camera, color, illumination, and scale changes. However, the algorithm, cannot discriminate doors from other large rectangular objects, such as bookshelves, cabinets, and cupboards (Munoz-Salinas et al., 2004; Tian et al., 2013). Besides, this method failed in detecting partially occluded doors (Shalaby et al., 2014). Probabilistic approach is another strategy explored by Murillo et al. (Murillo, Kosecka, Guerrero, & Sagues, 2008), which defined the likelihood of various features for generated door hypotheses, sowing a good performance, but for controlled environments. (Borgsen et al., 2014) used a RGB camera combining with depth information from laser sensor to identify basic structural door features. They extracted this information from point clouds and associate Gaussian probabilities to create an overall probability measurement of door recognition. Their approach, however, demands high computer hardware and processing time in identification of points cloud and if the distance of the door increases, the used camera presented low resolution making difficult the door detection process.

The massive number of studies which used the geometric models enabled to select door features to apply machine learning (ML) as AdaBoot classifier and Self-Organizing Map Networks. (Chen & Birchfield, 2008) and (Blaich & Bittel, 2010) trained an AdaBoost classifier to

Table 1
Studies in the SoA focused on computer vision-based solutions for door detection.

Study	Purpose	Sensor	Method	Dataset	Results
(Tian et al., 2013)	Blinded persons	RGB Camera	Canny edge Hough transform Matching edges	221 images (recorded)	Sen. = 89.5 %
(Shalaby et al., 2014)	Assistive robots	RGB Camera	Canny edge Hough transform Matching edges	210 images	Sen. = 78.2 %
(Dai et al., 2013)	Robot Navigation	RGB-D Camera	Hough transform	–	–
(Derry & Argall, 2013)	Wheelchair	RGB-D Camera	Hough transform	–	Sen. = 90 %
(Sekkal et al., 2013)	Wheelchair	RGB-D Camera	Geometric features 2D Edge tracking	3414 images (recorded)	Prec. = 82 %
(Fernández-Caramés et al., 2014)	Robot Navigation	RGB Camera Laser	Canny edge	Video recorded	Sen. = 77.33 %
(Borgsen et al., 2014)	Robot Navigation	RGB Camera Laser	Geometric features extraction	Video recorded	Sen. = 80 %
(W. Chen et al., 2014)	Robot Navigation	RGB Camera	Convolutional Neural Network	20 images (recorded)	Error ratio = 2.82 %
(Banerjee et al., 2015)	Humanoid Robot	RGB-D Camera	Geometric features RANSAC algorithm	–	–
(Yuan et al., 2016)	Robot Navigation	RGB-D Camera	Geometric features	Video recorded	–
(Kakillioglu et al., 2016)	Robot Navigation	RGB-D Camera	RANSAC algorithm	–	–
(He & Zhu, 2017)	Robot Navigation	RGB Camera	Line segment detector Geometric measurements	142 images (241)	Prec. = 89.4 %
(Llopart et al., 2017)	Robot Navigation	RGB-D Camera	CNN	500 images (recorded)	Sen. = 90 %
(Quintana et al., 2018)	Robot Navigation	RGB Camera Laser	Color-depth discontinuities analysis	35 images (recorded)	Prec. = 98.6 %
(Ramoa et al., 2020)	Robot Navigation	RGB-D Camera	FastFCN HarDNet PointNet classifier	1246 images (recorded)	Acc. = 90.9 %
(Spournias et al., 2020)	Robot Navigation	RGB Camera Laser	Faster-RCNN	8 images	–
(Othman & Rad, 2020)	Humanoid Robot	RGB Camera	DenseNet	SRIN Dataset	Prec. = 97.46 %
(Lecrosnier et al., 2021)	Wheelchair	RGB-D Camera	YOLOV3 Neural network	866 images (MCIndoor20000 and ESIGELEC dataset)	Prec. = 90 % Recall = 80 %

Sen: Sensitivity; Prec: Precision.

detect doors by combining the features of pairs of vertical lines, concavity, gap between the door and floor, color, texture, kick plate, and vanishing point. However, some of these features (*e.g.,* a perceptible gap below a door and the floor and a kick plate) are not always present in different instances of a door. Still in these field of ML tools, (Mahmood & Kunwar, 2011) employed feature-based classification and used SOM Network, which relies on the vertical gap between the door and the floor. So, the camera should be fixed at a low position, and cannot detect opened doors.

By adding depth information to typical 2D input data, alternative methods emerged based on the use of 3D point cloud data to detect and differentiate doors from walls using Random Sample Consensus (RANSAC) estimator, as in (Banerjee et al., 2015) and (Souto, Castro, Gonçalves, & Nascimento, 2017). Additionally, authors in (Kakillioglu et al., 2016) explored the effect of apply RANSAC algorithm with slicing techniques and employed an Aggregate Channel Features algorithm to identify specifically gaps inside doors, which are computationally efficient for real-time applications. However, this method only detected open doors. More recently, (Ramoa et al., 2020) proposed two other methods based on 3D input data to differentiate in real-time between open, closed and semi-open doors. They used PointNet Classifier and semantic segmentation algorithms FastFCN and FC-HarDNet. Although these studies presented good performance and provide depth information about object detection, they typically used non-miniaturized RGBD cameras (such ash, Kinect, PrimeSense, RealSense, Astro Orbec) or combine the use of RGB cameras with laser or sonar sensors. Their use could be feasibility for indoor autonomous robots (assistive and social domotics, humanoids or robotic wheelchairs) but giving their size and weight are not suitable for portable and wearable devices. These systems require high computer costs in terms of processing time, memory, and power-supply.

With the development of pattern recognition and artificial intelligence (AI) techniques, novel algorithms were proposed as a way to replicate the learning process and adaptability of human beings to unknown and dynamically changing scenario whilst, simultaneously, reducing computational time (Llopart et al., 2017). Thus, DL started to be applied in object detection domain and are among those giving the best performances in all methods (Chen et al., 2014; Lecrosnier et al., 2021; Llopart et al., 2017; Othman & Rad, 2020; Spournias et al., 2020). Deep convolution neural networks can accommodate certain degree of transformation, deformation, and illumination variation. An example of such behavior is the Convolutional Neural Network (CNN), which have been widely used in object detection and in the field of door detection by (W. Chen et al., 2014; Llopart et al., 2017; Othman & Rad, 2020; Spournias et al., 2020) for door detection. This network involves two steps: firstly, a judgment is made to check whether a door exists in the image and secondly, it can locate the door (W. Chen et al., 2014). Other explored DL-based technique was the one-stage method, which perform the object localization and object classification in a single network, the YOLOv3 network (Lecrosnier et al., 2021).

(Chen et al., 2014) used twenty images with the same features and by applying different image processing, increased their dataset up to 20,500 images in order to train a CNN. (Llopart et al., 2017) used a CNN to extract a region of interest from an image corresponding to a door or cabinet, combined with several method to extract point cloud data to detect handles inside the region of interest. Also, (Othman & Rad, 2020),

trained a CNN in SRIN dataset, a specifically dataset of images designed for indoor settings and images from short robots such as Nao robot, which was used to acquire the 2D images from its top camera. (Spournias et al., 2020) introduced the use of TF API object detection to implement a Faster CNN, which belong to the Comum Objects in Context (COCO) pre-trained models. They used a RGB camera to detect open and closed doors for Turtlebot 2 robotic platform, an ambient assisting living robot. On the other hand, authors in Lecrosnier et al. (Lecrosnier et al., 2021) adapted a YOLOv3 object detection algorithm to detect that their robotic wheelchair was in front of a door and by using an RGB-D camera, they used depth data for 3D door tracking. Although the overall high performance of these studies, they relied on the use of non-miniaturized cameras (Microsoft Kinect (Llopart et al., 2017), Naos' camera (Othman & Rad, 2020), 3D Orbec (Spournias et al., 2020) and Intel RealSense (Lecrosnier et al., 2021), which demands high power computational and power-supply and are not viable for wearable devices.

2.1. Highlights

State-of-the-art survey in computer vision-based solutions for door detection enabled to verify that the geometric models, probabilistic approaches, and the fuzzy logic methods required that all doors present the same features (shape, frame, concavity, gap between the door and floor, color, texture, kick plate, handles or typical associated text). This environment-controlled requirement was also a limitation for studies that applied ML classifiers (AdaBoost and SOM) which relied on these door features extraction and did not lead with occluded doors. These gaps appeared to be overcame with DL-techniques. However, current solutions used complex cameras, required high computer performance, and were implemented in indoor navigation robots. Although we verified that DL-based algorithms enabled to achieve the adaptability required for object detection models, current solutions are not reproducible for wearable and portable devices. Besides, we observed that no other study was implemented in a low-power computer device as an RPi and did not address PD problematic. Thus, this research followed a systematic approach to overcome the identified gaps in SoA solutions and customize our solution for PD end-users, by: (i) using a miniaturized, portable, and easily wearable camera, instead of the traditional and more complex Kinect, PrimeSense, RealSense or Astro Orbec cameras (Banerjee et al., 2015; Dai et al., 2013; Derry & Argall, 2013; Fernández-Caramés et al., 2014; Lecrosnier et al., 2021; Llopart et al., 2017; Othman & Rad, 2020; Quintana et al., 2018; Ramoa et al., 2020; Spournias et al., 2020; Tian et al., 2013; Yuan et al., 2016); (ii) implementing a DL-based model for door detection addressing PD problematic instead of indoor robot navigation; and (iii) develop a camera-based system to be integrated into a wearable device using a lighter, portable and low-power computer device, an RPi.

3. Deep learning for object detection *vs* system requirements

(Adrian Rosebrock, 2017) described DL as the latest incarnation of neural networks, but much faster, making use of specialized hardware and with more available training data. Instead of hand-defining a set of rules and algorithms to extract features from an image as in traditional ML algorithm, these features are automatically learned from the training process, specifically by CNNs (Adrian Rosebrock, 2017; Amit et al., 2020). One-stage CNNs methods perform the object location and classification with a single network, *i.e.,* they consider object detection as a regression or classification problem, being ideally methods for solutions with real-time constraints [7]. These CNNs mainly include MultiBox, Attention Net, You Only Look Once (YOLO) algorithms, Single Shot MultiBox Detector (SSD), among others (Adrian Rosebrock, 2017). These algorithms combine methods for locating and classifying regions of interest, thus avoiding resampling of pixels and features extracted from the image for each bounding box.

SSD models present more layers which helps the network to better detect objects in multiple scales and SSD bounding boxes can wrap around the objects in a tighter, more accurate fashion (Adrian Rosebrock, 2017). SSD structure includes a backbone network (*e.g.*, VGG or MobileNet, among others) and additional several features' layers to the end of the network. The additional feature layers are responsible for predicting the offsets to default boxes with different scales and aspect ratios and their associated confidences [8]. The network is trained with a weighted sum of localization loss and confidence loss, while final detection results are obtained by conducting non-maximum suppression on multiscale refined bounding boxes. From more detailed and careful dataset prepossessing (*e.g.*, by data augmentation), SSD significantly outperform the Faster R-CNN in terms of accuracy on PASCAL VOC and COCO datasets while being three times faster (Adrian Rosebrock, 2017; Amit et al., 2020). Further, SSD can provide models more effective and accurate than YOLO approach (Adrian Rosebrock, 2017). When comparing feature extraction layers, MobileNet reached an accuracy identical to VGG-16 on ImageNet with less cost (Amit et al., 2020).

Training from scratch an entire convolution network can require implementing by hand many network components, such as several custom layers and loss functions, besides to be time consuming and needs big datasets (Adrian Rosebrock, 2017; Amit et al., 2020). Given these fragilities, transfer learning is often applied to overcome these problems using pre-trained models for specific tasks in higher datasets, such as ImageNet or COCO, which are available in open-source libraries (Adrian Rosebrock, 2017). It is possible to leverage knowledge (features, weights, *etc*.) from previously trained models for training newer models even for dataset with less data. TF Object Detection API (TFOD-API) contains multiple out-of-the-boxes object detection structures like MobileNet-SSD which can be re-trained and tuned for new specific tasks.

3.1. How the proposed object detection model achieved the functional requirements to be integrated in the wearable biofeedback system?

Functional requirements were identified for the proposed object detection model, as indicated on Table 2: (1) time-efficiency, running at least ~0.5fps, considering mean step time of ~ 0.567sec measured in (Branquinho, Goncalves, Pinto, Rodrigues, & Santos, 2021) with ten healthy persons; (2) good performance in objects detection, measured by an acceptable F_1score of at least 0.85 as indicated on (Adrian Rosebrock, 2017); and (3) portability, by using low power consumption and miniaturized electronic devices aiming to be easily integrated on the wearable biofeedback device. Given our application will be based on spatiotemporal data through video sequences, we choose to use a pre-trained SSD-model. Also, to further tackle the practical limitations of running in real-time a high resource and power-consuming neural networks in a low-power device, we choose to use a MobileNet as backbone network of the SSD architecture. Thus, this research used a pre-trained MobileNet-SSD model using TFOD-API, which was pre-trained in COCO 2017 dataset and obtained a speed of 19 ms (~52.63fps) and 20.2 % mean average precision (Rathod, Joglekar, & Lu, 2017), which for us is a good trade-off between time-consumption for real-time performance and model precision. Innovatively, to the best knowledge of the authors, this DL-based approach was still not used for door detection, especially in the PD field, and used to run in a lower-power device, an RPi.

Table 2
Functional requirements identified for the proposed object detection model and respective key-indicator metrics of how HW-SW solutions achieved the identified requirements.

Requirement	Key-indicators	HW-SW solutions
Time-efficiency	≥0.5fps	Pre-trained MobileNet-SSD
High performance	F1-score ≥ 0.85	
Real-time application in a portable device	Low power consumption	RPi
	Miniaturized components	Camera RPi module
	Faster and lighter computing power model	Converted final model with TF-Lite

4. Methods: End-to-end door detection model

We designed an end-to-end pipeline for real-time door detection DL-based model for RPi, as shown in Fig. 1. In this section, we firstly present the dataset preparation and labeling procedure. Next, we focus on the DL approach, highlighting the adaptations implemented on the pre-trained MobileNet-SSD model. Following, we present the outcomes from the evaluation model phase. We finish this section with a sub-section focusing the conversion of the trained model TF-Lite version to be used in RPi.

4.1. Dataset preparation

The dataset was comprised by 912 images of doors with different sizes, angles, colors, states (closed/open/semi-open) and in several environments (*e.g.*, home, hospitals, or shopping) recorded by us and taken from google images. Given object detection models require supervised learning (needs a ground-truth), we created our training dataset by labeling the doors in the gathered images using the *LabelImg* tool. The procedure consisted in drawing a bounding box (SSD models detection output) around the exact location of the object to detect, as depicted in Fig. 2. To increase the model performance, besides to use images with different doors (*e.g.*, sizes, angles, colors, states and in several environments), we used images where doors were placed with similar objects, as closet, windows, fridges, or rectangular objects of different scenarios, as depicted in Fig. 2. The file format for the bounding box and labeling information was saved in Extensible Markup Language (XML) aiming the format to be accepted in MobileNet-SSD model. After this process, the input dataset of the object detection models included XML files of door images with the bounding box position and labeling information. The ratio of the training data to the test (unseen) data was 80:20 as in (Rathod et al., 2017) and from XML files, we generated a *record* files (*train.record* and *test.record*) to create all data to configure the model pipeline. Additionally, we created a label map file (*label_map.pbtxt)* which was used in training stage to indicate the number and ID of classes to be detected.

4.2. Training model pipeline configuration

A pre-trained MobileNet-SSD model from TFOD-API was used by configuring the model pipeline file. Fig. 3 depicts the used MobileNet-SSD v2 architecture, which was composed by 3 × 3 convolution layer, followed by 17 bottleneck residual blocks and additional 3 × 3 convolution layers, as indicated in Table 3.

The pipeline was divided into several essential structures that are responsible for dataset preparation, defining the model, training, and evaluation process. This pipeline configurations included some model parameters tuning: (1) definition of number of classes (one class); (2)

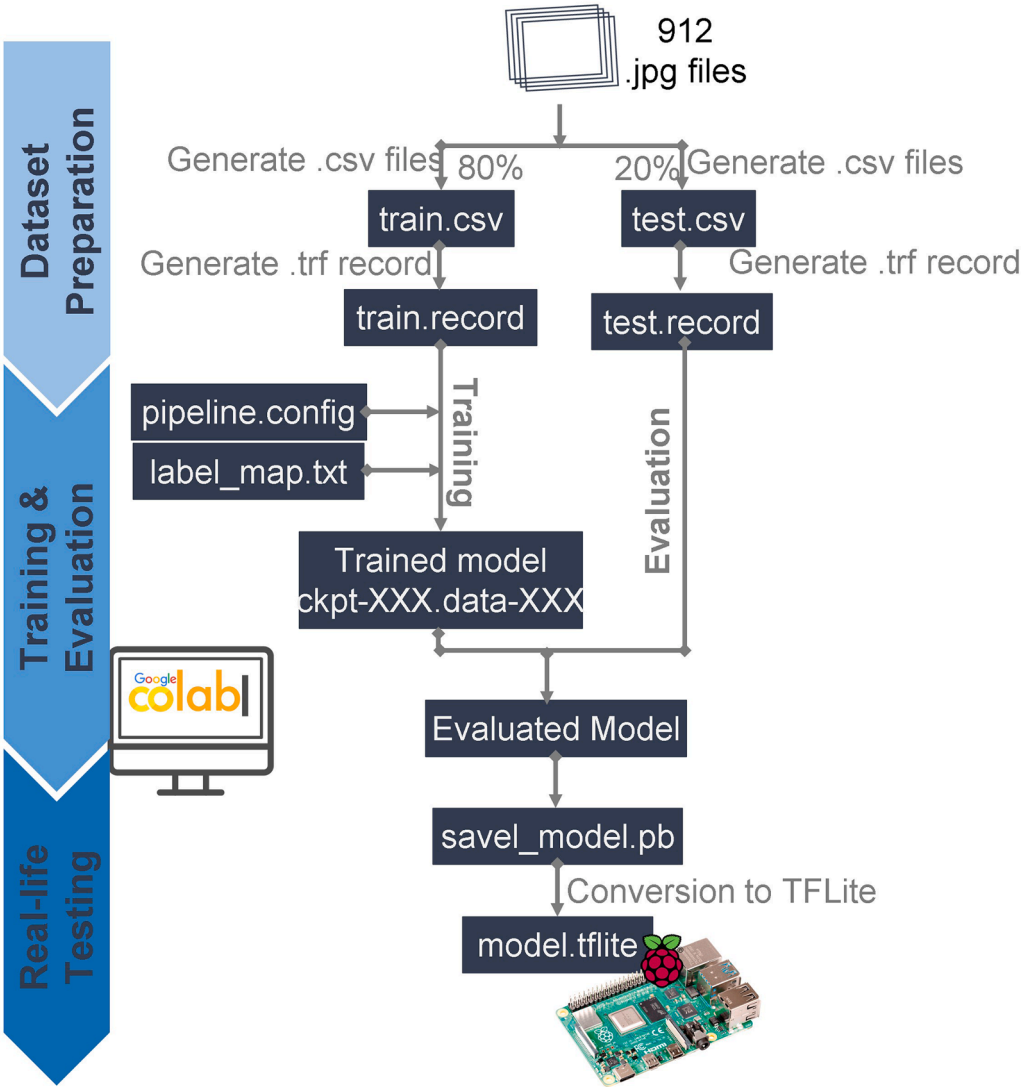

Fig. 1. End-to-end door detection model pipeline.

Fig. 2. Examples of doors' bounding boxes in our dataset. The dataset was comprised by 912 images of doors with different sizes, angles, colors, states (closed/open/semi-open) and in several environments (*e.g.*, home, hospitals, or shopping). Further, we used images where doors were placed with similar objects, as closet, fridges or rectangular objects of home scenarios.

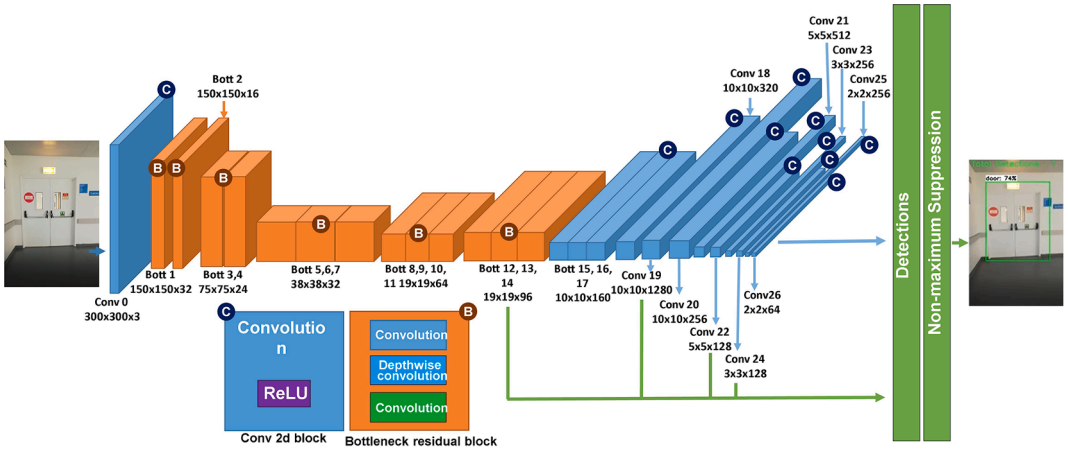

Fig. 3. Used MobileNet-SSD v2 architecture.

indication of label map and train/test RECORD files (dataset input) directories; (3) addition of data augmentation methods: im-age normalization, random adjust brightness/contrast/saturation/color, random jitter boxes and random resize/patch/crop) aiming to achieve more variations in data input; (4) definition of batch size value to 16; (5) use of Adam optimizer algorithm, instead of the pre-defined momentum optimizer, since it is suitable for large data, as images, can handle sparse gradients on noisy problems, be more stable and learns faster than other traditional optimizers; and (6) after tome tests and tuning, the learning rate was set to 0,025. Given this model was pre-trained in 300x300 COCO images, the pipeline configuration file also defines a first resize stage to guarantee 300x300 input data and improve detection speed. Model was trained in Google Colab using a Nvidia GPU server.

4.3. Performance of the DL-based object detection model

To evaluate the performance of the applied object detection model, we used the unknown data of the testing dataset to assess the results of the training model. We used the deductible outcomes from the standard statistical measures of Intersection over Union (IoU) used in DL-based models of object detection: precision, recall, and F1-score. When the estimated IoU value is higher than 0.5, the classified results of object

Table 3
MobileNet-SSD v2 structure layers and parameters.

Block name/number	Input	t	c	n	s
Conv 0	300 × 300 × 3		32		
Bott 1	150 × 150 × 32	–	16	1	2
Bott 2	150 × 150 × 16	1	24	1	1
Bott 3 4	75 × 75 × 24	6	24 32	1	1
Bott 5 6 7	38 × 38 × 32	6	32 32 64	2	1
Bott 8 9 10 11	19 × 19 × 64	6	64 64 64 96	3	1
Bott 12 13 14	19 × 19 × 96	6	96 96 160	4	1
Bott 15 16 17	10 × 10 × 160	6	160 160 320	3	1
Conv 18	10 × 10 × 320	–	1280	3	–
Conv 19	10 × 10 × 280	–	256	1	–
Conv 20	10 × 10 × 256	–	512	1	–
Conv 21	5 × 5 × 512	–	128	1	–
Conv 22	5 × 5 × 128	–	256	1	–
Conv 23	3 × 3 × 256	–	128	1	–
Conv 24	3 × 3 × 168	–	128	1	–
Conv 25	2 × 2 × 256	–	64	1	–
Conv 26	2 × 2 × 64	–	128	1	–

t: expansion factor, c: number of output channels, n: repeating number, s: stride.

detection are defined as true positive (TP), while if the vale is below to 0.5, a False positive (FP) is considered. False negative (FN) means that the predicting results should be positive, but the models have performed incorrect detection. Fig. 4 depicts these classified results. Based on these classified results (TP, FP and FN), the indicator of precision, recall, and F1-score can be further calculated, as indicated on equations 2–4, respectively. Precision evaluates the ability of the model with negative datasets, while recall represents the recognition ability with positive datasets. In computer vision field, these values are analyzed as mean average, meaning that each value is estimated for each class (on, our case, it was considered one class - doors): a) mean average precision (mAP); and b) mean average recall (mAR). F1-score establishes a balance between precision and recall, highlighting the robustness of the algorithm.

$$\text{Precision (mAP)} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{All detections}} \tag{1}$$

$$\text{Recall (mAR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{All truths}} \tag{2}$$

$$F_1\ score = 2 \cdot \left(\frac{Precision \cdot Recall}{Precision + Recall} \right) \tag{3}$$

4.4. Object detection model in RPi

TF-Lite is a massive improvement from standard TF installation libraries which significatively improves the performance of TF library. It is optimized to run on mobile and other low-resource computer devices such as the RPi (Adrian Rosebrock, 2017). After the final model has been trained and evaluated, before to be transferred for the end-device, the model was converted to a TF-Lite version, being much lighter and faster for real-time implementations as required for our system. To analyze these improvements, after we transferred the model to the RPi and configure the RPi camera module, we analyzed the time that algorithm took loading the model and to perform a real-time door detection (camera acquisition – object detection, in fps). We also accomplished a real-life testing assessment with video images taken in the final applicability scenario, with the final wearable biofeedback device. Three main systems of the wearable biofeedback device were used in this research, as indicated in Fig. 5.a: (1) a Sensory System - an RPI camera module; (2) a Processing System - an RPi 4B + model; and (3) a Power Supply System - an Anker® powerbank. All electronic devices are integrated into a 3D-printing box fixed on a waistband providing the required portability. In Fig. 5.b, it is also depicted how each system contributes to providing biofeedback for patients with PD. The Sensory System (RPi camera module) captures video frame sequences and sends each frame captured to the Processing System (RPi). The RPi establishes the interface with the camera module to receive each collected frame and runs in real-time the DL-based model (TF-Lite version) to detect the doors. If a door is detected, the model returns the position of the classified object on the input frame, but when a door is not detected the model returns no objects. When a door is detected, this information is sent to a fourth system included on the wearable biofeedback device, the Actuation System. This system measures the distance between the patient and the detected door and provides visual sensory cues. The Anker® powerbank energetically supplies the electronic components for an autonomy of ~8 h."

5. Results

5.1. Performance of the DL-based object detection model

Table 4 presents the performance of the trained models during the fine-tuning phase (Train ID 1 and 2), using the NVIDIA GPU server from Google Colab. It is also presented the performance of the obtained final model (Train ID 3), which was used to be converted to a TF-Lite model aiming to be run in the RPi. For each trained model it is highlighted some model parameters (batch size, use of data augmentation processing, optimizer method and learning rate) which have been tuned to achieve the final model performance. Training time varied with batch size values, use of data augmentation techniques and learning rate, being achieved a lower training time when the batch size decreased to 16 as also the learning rate to 0,025 with Adam optimizer. Ultimate trained model precision showed a higher mAP (@0.5IoU) of 97,2%. A higher mAP indicates that the door label is classified more precisely. For further inspection of the trained model, we analyzed recall (mAR) and F1-score, also indicated on Table 4. Significant values of recall means that the model has a positive ability to find all door labels. We achieved an acceptable ~80 % probability that the model truly classifies as a true positive. Additionally, Table 4 provides F1-score values as a harmonic

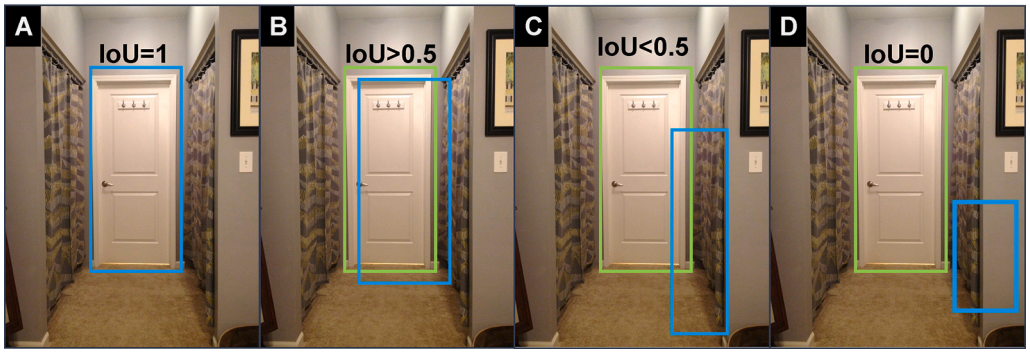

Fig. 4. Classified results. A: "Perfect" true positive (IoU = 1); B: True positive (IoU > 0.5); and C: False positive (IoU < 0.5). False negative (IoU = 0).

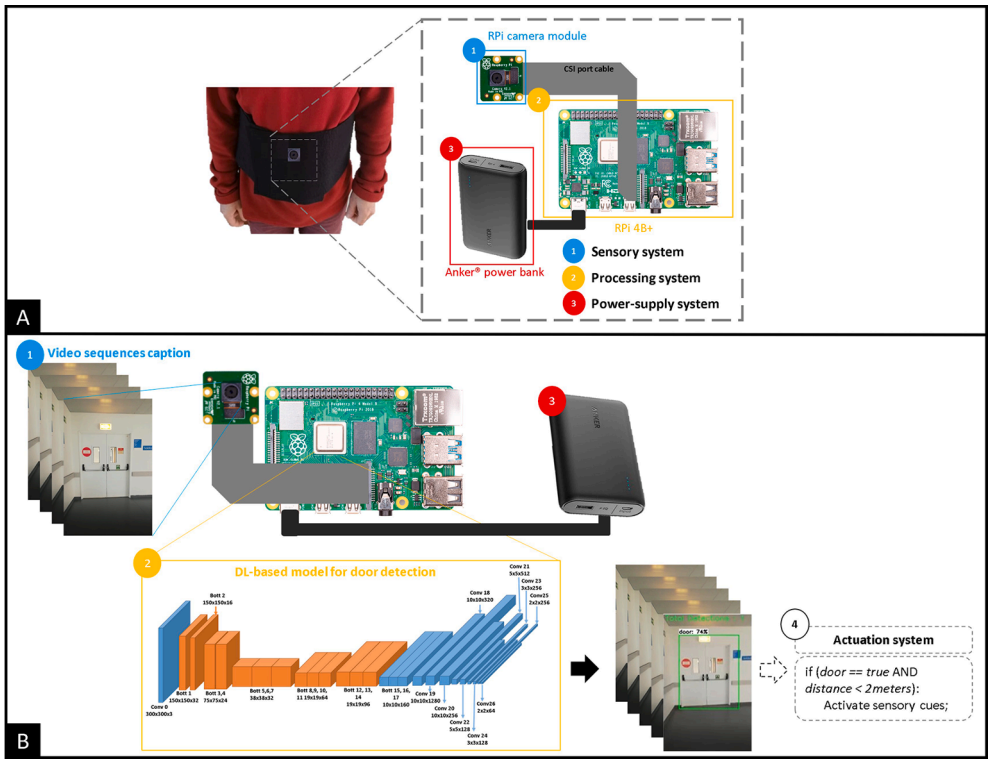

Fig. 5. A: Three main components used on this research from wearable biofeedback device: sensory, processing and power-supply systems integrated into a waistband; **B:** Scheme of how each system contributes to the wearable device be able to provide biofeedback for patients with PD. Sensory system is responsible for caption of videos sequence and send each frame captured to the processing system. The processing systems uses the received frame to supply and run the DL-based model to find if a door is detected. When a door is detected, this information is sent to a fourth system, an actuation system. Power supply system provides the energy required for the electronic devices.

Table 4
Performance of the trained models tuned with different models' parameters.

Train ID	Parameters				Performance			
	Batch size	Aug.	Optimizer	Learning rate	Training time [h]	Precision (mAP)	Recall (mAR)	F1-score
1	64	N	Momentum	0.8	~54	0,967	0,779	0,863
2	64	Y	Adam	0.8	~92	0,968	0,783	0,867
3	**16**	Y	**Adam**	**0.025**	**~14**	**0,972**	**0,789**	**0,869**

mean of precision and recall, showing a good performance of ~0,869, meaning that there is a good proportion of the model to categorize as a positive outcome and be an actual positive case.

5.2. Object detection model in RPi

Once obtained the final model, *i.e.*, after training and evaluating the DL-based model, we converted the model to a TF-Lite model. This step enabled to improve the computational power when running the obtained model in real-time, more precisely on the portable device (RPi). It took over 1 min (~58sec) to load the non-converted model and less than one second (~0.00296sec) to load the final TF-Lite model, as indicated on Fig. 6.

Fig. 7 presents the outcomes of the real-life testing assessment with video images taken in the final applicability scenario, using the RPi integrated into a wearable device (instrumented waistband) running the DL-based trained converted TF-Lite model. The converted model takes less memory (6810 Kb to 5032 kb) than the non-converted model. This conversion presented a direct impact on real-time model running, where we could obtain ~2.87 fps, instead of the initial ~0.52 fps.

6. Discussion

A DL-based object detection model implemented in a RPi was described. We used precision to measure how accurate is the trained model for object detection, *i.e.*, the percentage of objects detected that were correct. We also assessed the recall to study how good is the model finding all the true positive objectives. Furthermore, we estimated the F1-score to establish a balance between the measured precision and recall.

In fine-tuning phase, we performed three training models as indicated on Table 4. We started by measuring the performance of a model training (Train ID 1) using the default pre-trained model configurations (batch size = 64, optimizer = momentum, absence of data augmentation and learning rate = 0.8). Results from Train ID 1 already exhibit a good model performance (mAP = 96,8%; mAR=0,779 %; and F1-score = 0,863). However, aiming to improve the model performance, we accomplished a second training adding data augmentation methodologies and changing the optimizer method to the Adam optimizer. Data augmentation is a popular technique largely used to enhance the training of CNNs. As a matter of fact, in (Hernández-García & König,

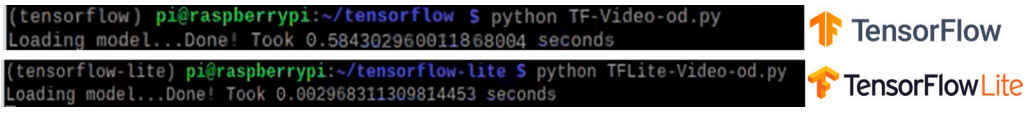

Fig. 6. Object detection model performance regarding the time required to load the developed model with TF and TF-Lite environment.

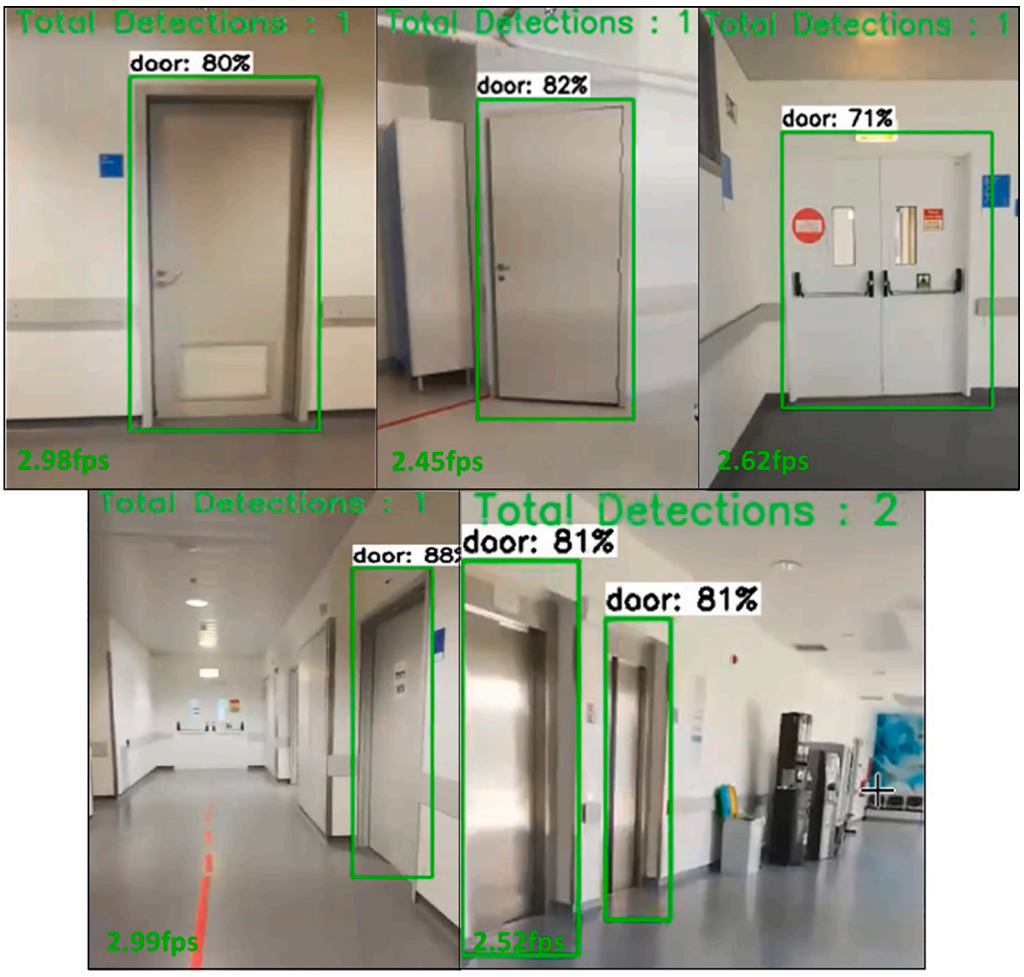

Fig. 7. Object detection model performance regarding real-life results with doors with different sizes, angles, and types of doors. The detected doors are bounded by the green rectangle; it is highlighted the total number of doors detected, fps used to capture the image & classify, and the percentage of confidence of classification.

2018) it was observed that networks which trained with data augmentation more easily adapt to different architectures and amount of training data, as opposed to weight decay and dropout techniques in improving model performance, which require specific fine-tuning of their hyperparameters. In computer vision, data augmentation can be obtained by altering image features like brightness, color, hue, orientation, or cropping. Our approach included variations on images brightness, contrast, saturation, color, resizing, cropping, gaussian patch and jitter boxes. Indeed, this approach enabled to improve model sensitivity, *i.e.,* true positive rate (*mAR*) from 77,9% to 78,3% (Train ID 2), which improved correct detections as shown in Fig. 7, where doors may be confused with similar shape objects as doors of closets or fridges. In Train ID 2, we also changed the pre-defined momentum optimizer to Adam, given its ability of being computationally efficient by requiring little memory and be faster in learning converging. Further, with Adam optimizer, we achieved a training more computationally efficient since it was required little memory.

Another parameter that needed to be tuned was the batch size. Using smaller batch sizes has been empirically shown to have faster convergence to acceptable performances, as indicated in (Kandel & Castelli, 2020; Masters & Luschi, 2018; Radiuk, 2018). This is intuitively explained by the fact that smaller batch sizes allow the model to start learning before having to look at all data, converging faster and requiring less time to train, as occurred in Train ID 3 (~14 h) (Hernández-García & König, 2018; Masters & Luschi, 2018). Despite the use of a low batch size enabled to achieve a faster model converging, requiring less memory, the global optima convergence could not be guaranteed (Kandel & Castelli, 2020). To guarantee that the reduction in batch size value did not affect the model performance, besides using data augmentation, we decreased the learning rate to provide more stability to the learning process and model generalization performance as referred in (Masters & Luschi, 2018) and (Kandel & Castelli, 2020).

Regarding the final trained model (Train ID3), which was used to be converted with TF-Lite to run on RPi, the results showed a considerable *mAP* (@0.5IoU) of 97,2%, *mAR* of 78,9% and a good F1-score of 0,869. Although a point-by-point comparison was not done since different methods and datasets were employed, improved results were achieved with the proposed solution, comparatively to the most recent related SoA DL-based models (studies since 2020), as described in Table 5.

Most recent studies used FastFCN, DenseNet and YOLOV3 models, which are not indicated for our purpose, as following discussed. Similar to the selected DL method (MobileNet-SSD), FastFCN model used by (Ramoa et al., 2020) is also a pre-trained model with COCO 2017 available in TFOD-API for transfer learning. The FastFCN model achieved better mAP (37.7 %) in COCO 2017 dataset training than the MobileNet-SSD (20.2 %). However, its real-time performance falls far short (206 ms), when compared to the MobileNet-SSD (19 ms), being FastFCN model not feasible to address our time-efficiency requirement. Also, as indicated by (Adrian Rosebrock, 2017; Amit et al., 2020), from a more detailed and careful dataset preprocessing, MobileNet-SSD can significantly outperform FastFCN, as verified in this research.

Although (Othman & Rad, 2020) with DenseNet model presented a precision similar to the obtained in this research, DenseNet is indicated for image classification not performing image location (object detection)

Table 5
Current SoA of DL-based models for door detection.

Study	Purpose	Camera module	Method	Dataset	Results
(Ramoa et al., 2020)	Navigation	RealSense D435	FastFCN	1246 images	Acc = 90.9 %
(Othman & Rad, 2020)	Humanoid Robot	Nao's monocular cameras	DenseNet	SRIN Dataset	Prec = 97.46 %
(Lecrosnier et al., 2021)	Wheelchair	RealSense D435 RealSense T265	YOLOV3	866 images	Prec = 90 % Recall = 79 %
Our solution	Motor assistance PD	RGB Camera	MobileNet-SSD	912 images	Prec = 97.2 % Recall = 78.9 % F1-score = 0,869

Acc: accuracy; Prec: precision.

(Yin, Hong, Zheng, Chen, & Deng, 2022). Thus, for our purpose of developing a wearable biofeedback device for PD motor assistance, that includes a DL model for object detection, aiming to identify if and where an object (door) is recognized, MobileNet-SSD was a more ideal solution. Further, SSD models can provide more effective and accurate approaches than YOLO as indicated by (Amit et al., 2020) and observed through an overall comparison to our solution performance (precision of 97.2 %) with that of (Lecrosnier et al., 2021) study (precision of 90 %).

We also verified from Table 5 that (Lecrosnier et al., 2021; Othman & Rad, 2020; Ramoa et al., 2020) used more complex and non-miniaturized cameras as RealSense D435 in (Ramoa et al., 2020), Nao humanoid monocular cameras in (Othman & Rad, 2020) and RealSense D435 and RealSense T265 cameras in (Lecrosnier et al., 2021). Besides the use of these non-miniaturized cameras may compromise the integration of the system into the wearable device, they require more powerful computer devices.

Summing up, we believe that we contribute to the related SoA with a new DL-based model customized to our purpose: (1) able to provide doors detection with significant accuracy, sensitivity, and time-efficiency; (2) using lighter, easily portable, and miniaturized camera module; and (3) capable to run in a low power device with a good performance.

Patients with PD tend to walk slowly, as observed in (Branquinho et al., 2021), where, *e.g.,* 20 patients performed a mean step duration of ~0.601sec when walking 10 m. When testing the model in the RPi integrated on the wearable device, it was observed that by using TF-Lite environment, in real-time detection, we achieved a ~2.87 fps to capture and identify doors, which is a time-effective performance to detect step changes over real-life scenarios Fig. 6. Indeed, considering the outcomes measured in (Branquinho et al., 2021), with our solution we have ~2frames in a step time of 0.601sec to identify that the patient is in front of a door and deliver on-demand the proprioceptive cue. Also, the DL model was accurate and efficient in detecting doors in real-life scenarios, as observed in Fig. 7. Based on these findings, the developed method achieved the pre-identified functional requirements indicated on Table 2.

7. Conclusions

This study aimed to implement and test a DL-based object detection application, implemented in a RPi using a camera module. From here, it is expected to contribute for PD scope, aiming to develop a wearable device for delivering proprioceptive cues by real-time detection of patients' environment using the implemented DL model. It was used transfer-learning concepts to train a MobileNet-SSD in TF environment. Since, the model was then integrated in a RPi, the model was converted to a faster and lighter computing power model using TF-Lite environment. The model showed a significant *mAP* of 97,2%, *mAR* of 78,9% and a good F1-score of 0,869, overcoming the related SoA DL-based models for door detection. When testing the model into RPi integrated on the wearable device, it was observed that by using TF-Lite environment, the DL-model showed to be temporally efficient (~2.87fps) to detect with accuracy the doors in real-life scenarios.

This manuscript described the first advances on the development of a wearable high-tech to provide context awareness biofeedback on PD. Thus, future challenges will cover to integrate the developed model with customized sensory cues. Also, it is required to validate the device with end-users in home scenarios with quotidian motor tasks (*e.g.*, passage through doors between kitchen and living rooms).

CRediT authorship contribution statement

Helena R. Gonçalves: Conceptualization, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Cristina P. Santos:** Conceptualization, Formal analysis, Investigation, Methodology, Software, Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by FCT National Funds, under the National support to R&D units grant, through the reference project UIDB/04436/2020 and UIDP/04436/2020, and under the Reference Scholarship under grant SFRH/BD/136569/2018.

References

Adrian Rosebrock. (2017). Deep Learning for Computer Vision with Python: Practitioner Bundle. In Pyimagesearch (Ed.), *Deep Learning for Computer Vision with Python* (1.2.1, Vol. 53, Issue 9).

Amit, Y., Felzenszwalb, P., & Girshick, R. (2020). Object detection. *Computer Vision, 1–9*. https://doi.org/10.1007/978-3-030-03243-2_660-1

Banerjee, N., Long, X., Du, R., Polido, F., Feng, S., Atkeson, C. G., Gennert, M., & Padir, T. (2015). Human-supervised control of the ATLAS humanoid robot for traversing doors. *IEEE-RAS International Conference on Humanoid Robots, 2015-Decem*, 722–729. 10.1109/HUMANOIDS.2015.7363442.

Blaich, M., & Bittel, O. (2010). Real-time door detection based on AdaBoost learning algorithm. *Communications in Computer and Information Science*, *82 CCIS*(November), 61–73. 10.1007/978-3-642-16370-8_6.

Borgsen, S. M. Z., Schöpfer, M., Ziegler, L., & Wachsmuth, S. (2014). Automated door detection with a 3D-sensor. *Proceedings - Conference on Computer and Robot Vision, CRV, 2014*, 276–282. https://doi.org/10.1109/CRV.2014.44

Branquinho, A., Goncalves, H. R., Pinto, J. F., Rodrigues, A. M., & Santos, C. P. (2021). Wearable gait Analysis LAB as a biomarker of Parkinson's disease motor stages and Quality of life: A preliminary study. *2021 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2021*, *i*, 234–239. 10.1109/ICARSC52212.2021.9429770.

Chen, W., Qu, T., Zhou, Y., Weng, K., Wang, G., & Fu, G. (2014). Door recognition and deep learning algorithm for visual based robot navigation. *2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014*, 1793–1798. 10.1109/ROBIO.2014.7090595.

Chen, Z., & Birchfield, S. T. (2008). Visual detection of lintel-occluded doors from a single image. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, June*. 10.1109/CVPRW.2008.4563142.

Chong, R., Hyun Lee, K., Morgan, J., & Mehta, S. (2011). Closed-loop VR-based interaction to improve walking in Parkinson's disease. *Journal of Novel Physiotherapies, 1*(1), 1–7. https://doi.org/10.4172/2165-7025.1000101

Cowie, D., Limousin, P., Peters, A., Hariz, M., & Day, B. L. (2012). Doorway-provoked freezing of gait in Parkinson's disease. *Movement Disorders, 27*(4), 492–499. https://doi.org/10.1002/mds.23990

Dai, D., Jiang, G., Xin, J., Gao, X., Cui, L., Ou, Y., & Fu, G. (2013). Detecting, locating and crossing a door for a wide indoor surveillance robot. In *2013 IEEE International Conference on Robotics and Biomimetics*. https://doi.org/10.1109/ROBIO.2013.6739719

Delgado-Alvarado, M., Marano, M., Santurtún, A., Urtiaga-Gallano, A., Tordesillas-Gutierrez, D., & Infante, J. (2019). Nonpharmacological, nonsurgical treatments for freezing of gait in Parkinson's disease: A systematic review. *Movement Disorders, 1–11*. https://doi.org/10.1002/mds.27913

Derry, M., & Argall, B. (2013). Automated doorway detection for assistive shared-control wheelchairs. *Proceedings - IEEE International Conference on Robotics and Automation, 1254–1259*. https://doi.org/10.1109/ICRA.2013.6630732

Ehgoetz Martens, K. A., Pieruccini-Faria, F., & Almeida, Q. J. (2013). Could sensory mechanisms be a core factor that underlies freezing of gait in Parkinson's disease? *PLoS One, 8*(5). https://doi.org/10.1371/journal.pone.0062602

Fernández-Caramés, C., Moreno, V., Curto, B., Rodríguez-Aragón, J. F., & Serrano, F. J. (2014). A real-time door detection system for domestic robotic navigation. *Journal of Intelligent and Robotic Systems: Theory and Applications, 76*(1), 119–136. https://doi.org/10.1007/s10846-013-9984-6

Ginis, P., Nackaerts, E., Nieuwboer, A., & Heremans, E. (2018). Cueing for people with Parkinson's disease with freezing of gait: A narrative review of the state-of-the-art and novel perspectives. *Annals of Physical and Rehabilitation Medicine, 61*(6), 407–413. https://doi.org/10.1016/j.rehab.2017.08.002

Gómez-Jordana, L. I., Stafford, J., Peper, C. E., & Craig, C. M. (2018). Crossing virtual doors: A new method to study gait impairments and freezing of gait in Parkinson's disease. *Parkinson's Disease, 2018*. https://doi.org/10.1155/2018/2957427

Harrington, W., Greenberg, A., King, E., McNames, J., Holmstrom, L., Horak, F. B., & Mancini, M. (2016). Alleviating freezing of gait using phase-dependent tactile biofeedback. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. https://doi.org/10.1109/EMBC.2016.7592056

He, Z., & Zhu, M. (2017). Real-time door detection for indoor autonomous vehicle. *Ninth International Conference on Digital Image Processing (ICDIP 2017)*, *10420*(Icdip), 104200V. 10.1117/12.2281651.

Hensler, J., Blaich, M., & Bittel, O. (2010). Real-time door detection based on AdaBoost learning algorithm. *Communications in Computer and Information Science*, *82 CCIS* (November), 61–73. 10.1007/978-3-642-16370-8_6.

Hernández-García, A., & König, P. (2018). Further advantages of data augmentation on convolutional neural networks. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-030-01418-6_10

Ioannidou, A., Chatzilari, E., Nikolopoulos, S., & Kompatsiaris, I. (2017). Deep learning advances in computer vision with 3D data: A survey. *ACM Computing Surveys, 50*(2). https://doi.org/10.1145/3042064

Kakillioglu, B., Ozcan, K., & Velipasalar, S. (2016). Doorway detection for autonomous indoor navigation of unmanned vehicles. In *Proceedings - International Conference on Image Processing*. https://doi.org/10.1109/ICIP.2016.7533078

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express, 6*(4), 312–315. https://doi.org/10.1016/j.icte.2020.04.010

Krishna Sai, B. N., & Sasikala, T. (2019). Object Detection and Count of Objects in Image using Tensor Flow Object Detection API. In *Proceedings of the 2nd International Conference on Smart Systems and Inventive Technology*. https://doi.org/10.1109/ICSSIT46314.2019.8987942

Lecrosnier, L., Khemmar, R., Ragot, N., Decoux, B., Rossi, R., Kefi, N., & Ertaud, J. Y. (2021). Deep learning-based object detection, localisation and tracking for smart wheelchair healthcare mobility. *International Journal of Environmental Research and Public Health, 18*(1), 1–17. https://doi.org/10.3390/ijerph18010091

Lewis, S. J. G., & Barker, R. A. (2009). A pathophysiological model of freezing of gait in Parkinson's disease. *Parkinsonism and Related Disorders, 15*(5), 333–338. https://doi.org/10.1016/j.parkreldis.2008.08.006

Llopart, A., Ravn, O., & Andersen, N. A. (2017). Door and cabinet recognition using Convolutional Neural Nets and real-time method fusion for handle detection and grasping. In *2017 3Rd International Conference on Control, Automation and Robotics*. https://doi.org/10.1109/ICCAR.2017.7942676

Mahmood, F., & Kunwar, F. (2011). A self-organizing neural scheme for road detection in varied environments. *Proceedings of the International Joint Conference on Neural Networks, 60*(9), 3049–3054. https://doi.org/10.1109/IJCNN.2011.6033623

Mancini, M., Smulders, K., Harker, G., Stuart, S., & Nutt, J. G. (2018). Assessment of the ability of open- and closed-loop cueing to improve turning and freezing in people with Parkinson's disease. *Scientific Reports, 8*(1), 1–9. https://doi.org/10.1038/s41598-018-31156-4

Masters, D., & Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. *ArXiv*, 1–18. 10.48550/arXiv.1804.07612.

Munoz-Salinas, R., Aguirre, E., Garcia-Silvente, M., & Gonzalez, A. (2004). Door-detection using computer vision and fuzzy logic. *WSEAS Transactions on Systems, 3* (10), 3047–3052.

Murillo, A. C., Kosecka, J., Guerrero, J. J., & Sagues, C. (2008). *Visual door detection integrating appearance and shape cues*.

Muthukrishnan, N., Abbas, J. J., Shill, H. A., & Krishnamurthi, N. (2019). Cueing paradigms to improve gait and posture in parkinson's disease: A narrative review. *Sensors (Switzerland), 19*(24). https://doi.org/10.3390/s19245468

Othman, K. M., & Rad, A. B. (2020). A doorway detection and direction (3ds) system for social robots via a monocular camera. *Sensors (Switzerland)*, *20*(9). 10.3390/s20092477.

Pham, T. T., Moore, S. T., Lewis, S. J. G., Nguyen, D. N., Dutkiewicz, E., Fuglevand, A. J., … Leong, P. H. W. (2017). Freezing of gait detection in Parkinson's disease: a subject-independent detector using anomaly scores. *IEEE Transactions on Biomedical Engineering, 64*(11), 2719–2728. https://doi.org/10.1109/TBME.2017.2665438

Quintana, B., Prieto, S. A., Adán, A., & Bosché, F. (2018). Door detection in 3D coloured point clouds of indoor environments. *Automation in Construction*, *85*(October 2016), 146–166. 10.1016/j.autcon.2017.10.016.

Radiuk, P. M. (2018). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. *Information Technology and Management Science, 20*(1), 20–24. https://doi.org/10.1515/itms-2017-0003

Ramoa, J. G., Alexandre, L. A., & Mogo, S. (2020). Real-time 3D door detection and classification on a low-power device. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions*. https://doi.org/10.1109/ICARSC49921.2020.9096155

Rathod, V., Joglekar, S., & Lu, Z. (2017). *TensorFlow Models*. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md.

Sekkal, R., Pasteau, F., Babel, M., Brun, B., & Leplumey, I. (2013). Simple monocular door detection and tracking. *IEEE International Conference on Image Processing*, 3929–3933.

Shalaby, M. M., Salem, M. A. M., Khamis, A., & Melgani, F. (2014). Geometric model for vision-based door detection. In *Proceedings of 2014 9th IEEE International Conference on Computer Engineering and Systems*. https://doi.org/10.1109/ICCES.2014.7030925

Souto, L. A. V., Castro, A., Gonçalves, L. M. G., & Nascimento, T. P. (2017). Stairs and doors recognition as natural landmarks based on clouds of 3D edge-points from RGB-D sensors for mobile robot localization. *Sensors (Switzerland), 17*(8), 1–16. https://doi.org/10.3390/s17081824

Spournias, A., Antonopoulos, C., Keramidas, G., Voros, N., & Stojanovic, R. (2020). Enhancing Visual Recognition for Door Status Identification in AAL Robots via Machine Learning. In *2020 9Th Mediterranean Conference on Embedded Computing*. https://doi.org/10.1109/MECO49872.2020.9134108

Sweeney, D., Quinlan, L. R., Browne, P., Richardson, M., Meskell, P., & Ólaighin, G. (2019). A technological review of wearable cueing devices addressing freezing of gait in Parkinson's disease. *Sensors (Switzerland), 19*(6). https://doi.org/10.3390/s19061277

Tian, Y., Yang, X., & Arditi, A. (2010). Computer vision-based door detection for accessibility of unfamiliar environments to blind persons. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-14100-3_39

Tian, Y., Yang, X., Yi, C., & Arditi, A. (2013). Toward a computer vision-based wayfinding aid for blind persons to access unfamiliar indoor environments. *Machine Vision and Applications, 24*(3), 521–535. https://doi.org/10.1007/s00138-012-0431-7

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience, 2018*. https://doi.org/10.1155/2018/7068349

Yamagami, M., Imsdahl, S., Lindgren, K., Bellatin, O., Nhan, N., Burden, S. A., … Kelly, V. E. (2020). Effects of virtual reality environments on overground walking in people with Parkinson disease and freezing of gait. *Disability and Rehabilitation: Assistive Technology*, 1–8. https://doi.org/10.1080/17483107.2020.1842920

Yin, L., Hong, P., Zheng, G., Chen, H., & Deng, W. (2022). A novel image recognition method based on DenseNet and DPRN. *Applied Sciences (Switzerland), 12*(9). https://doi.org/10.3390/app12094232

Yuan, T. H., Hashim, F. H., Zaki, W. M. D. W., & Huddin, A. B. (2016). An automated 3D scanning algorithm using depth cameras for door detection. *Proceedings - 2015 International Electronics Symposium: Emerging Technology in Electronic and Information, IES 2015*, 58–61. 10.1109/ELECSYM.2015.7380814.

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2018). Object detection with deep learning: A review. *ArXiv, 30*(11), 3212–3232.